ELSEVIER Discrete Applied Mathematics 51 (1994) 219-232
DISCRETE APPLIED MATHEMATICS

Computing the jump number on semi-orders is polynomial

A. von Arnim[a], C. de la Higuera[b,*]

[a]Institut f. Informatik, University of Cologne, Cologne, Germany
[b]L.I.R.M.M. (D.I.F.), University of Montpellier 2 & CNRS UMC9928, Montpellier, France

(Received 1 October 1991; revised 28 December 1992)

Abstract

Semi-orders form a subclass of interval orders: they can be represented as sets of intervals of a given length. We first prove that semi-orders can be partitioned by serialization (or series decomposition) without loss of the jump number aspect. On non-serializable semi-orders all linear extensions contain never more than two consecutive bumps (maximal chains of length at most 3). We then give a "divide-and-conquer" argument proving that to solve this case all we need is to be able to compute the number of maximal chains of length at least 2. This can also be dealt with in polynomial time, allowing us to claim that computing the jump number is polynomial on semi-orders.

1. Introduction and notations

In this first section we will give our main definitions and recall different characterizations of interval and semi-orders.

In Section 2 we shall prove that after a decomposition routine, semi-orders have at most 2 consecutive bumps in a linear extension. We also prove, using a "divide-and-conquer" argument that computing polynomially the jump number can be done provided we can compute polynomially the jump number $s_2(P)$ for linear extensions with at most one consecutive bump. In Section 3 we provide an algorithm to calculate the jump number. Finally in Section 4 we provide a full example and discuss the complexity issues.

Interval orders have been thoroughly studied since the early seventies for their importance in the context of measurement theory (see [6, 7, 12]). They have come back into fashion lately for their possible applications to parallelism [8]. Semi-orders are a proper subclass of interval orders, restrictive but also with potential applications [15].

* Corresponding author. Email: delahiguera@lirrm.fr.

SSDI 0166-218X(93)E0077-C

Let $P = (X, <_P)$ denote a finite partially ordered set and recall the following notations:

$x \parallel_P y$ when x and y are not comparable in P,

y covers x if $x <_P y$ and there is no z such that $x <_P z$ and $z <_P y$,

$\mathrm{succ}(x) = \{y \in P \mid x <_P y\}$; the set of all successors of x,

$\mathrm{Succ} = \{\mathrm{succ}(x) \mid x \in P\}$; the set of all successor sets,

$\mathrm{pred}(x) = \{y \in P \mid y <_P x\}$; the set of all predecessors of x,

$\mathrm{Pred} = \{\mathrm{pred}(x) \mid x \in P\}$; the set of all predecessor sets.

The (*Hasse*) *diagram* of P is the directed graph with vertex set X and edges xy whenever y covers x in P. The direction of an edge xy is depicted by drawing y above x.

1.1. Interval and semi-orders

A poset P is called an *interval order* if it is representable by assigning a real interval $I_x = [a_x, b_x]$ to each element x in P, such that $x <_P y$ iff $b_x < a_y$. It is called a *semi-order* if there is a representation with intervals of the same length.

Hence two elements are incomparable if their corresponding intervals intersect. We shall also want the following condition to be fulfilled: if $\mathrm{succ}(x) = \mathrm{succ}(y)$ and $\mathrm{pred}(x) = \mathrm{pred}(y)$ then $I_x = I_y$. In such a case the following notations are well defined:

$x \equiv_P y$ if $I_x = I_y$.

$x \ll y$ if $a_x < a_y$.

This obviously yields that one of the following always holds:

(i) $x \ll y$ (ii) $y \ll x$ (iii) $x \equiv y$.

In the sequel P and X will be considered as the same set and the index P will be omitted whenever there is no ambiguity. Here are the last notations we will use:

$\mathrm{imsucc}(x) = \{y \mid y$ covers x and there exists no z s.t. $x \ll z$ and $z < y\}$,

$\mathrm{impred}(x) = \{y \mid x$ covers y and there exists no z s.t. $z \ll x$ and $y < z\}$,

xPy is the suborder of P on set $\{z \in P \mid x \ll z$ and $z \ll y\} \cup \{x, y\}$,

xP is the suborder of P on set $\{z \in P \mid x \ll z\} \cup \{x\}$,

Py is the suborder of P on set $\{z \in P \mid z \ll y\} \cup \{y\}$.

Let us expose some of these notions in Fig. 1.

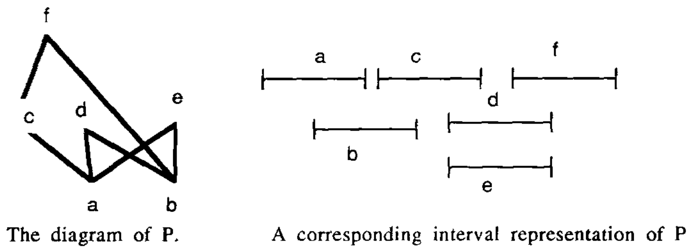


Fig. 1. $\mathrm{succ}(a) = \{c,d,e,f,g\}$; $\mathrm{pred}(c) = \{a\}$; $\mathrm{imsucc}(b) = \{d,e\}$; $\mathrm{impred}(e) = \{b\}$; $b \lessdot c$, $c \lessdot d, \ldots$; $d \equiv e$; $aPf = P$.

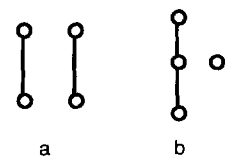


Fig. 2.

We now recall the classical characterizations of an interval order.

Theorem 1.1 (Fishburn [6]). *For a poset $P = (X, \leqslant)$ the following* 4 *statements are equivalent:*

(i) *P is an interval order.*

(ii) *P does not contain any subposet isomorphic to* $2 + 2$ (*Fig.* 2(*a*)).

(iii) *The maximal antichains of P can be linearly ordered such that, for every element x, the maximal antichains containing x occur consecutively.*

(iv) *The sets of predecessors* $\mathrm{pred}(x)$ (*resp.*, $\mathrm{succ}(x)$) *are totally ordered by inclusion.*

If one of the above properties is true then we furthermore have:

(v) $|\mathrm{Pred}| = |\mathrm{Succ}|$.

In the same way we can characterize semi-orders: $P = (X, \leqslant)$ is a semi-order iff P does not contain any subposet isomorphic to $2 + 2$ or $3 + 1$ (Fig. 2(b)).

1.2. The jump number problem

Let $\tau = x_1 \ldots x_n$ be a total ordering of the elements of P. τ is a *linear extension* of P if $x <_P y$ implies x is before y in τ ($x <_\tau y$).

Two consecutive elements of τ, x_i and x_{i+1} are separated by a *jump* (resp. *bump*) when x_i and x_{i+1} are not comparable in P (resp. $x_i <_P x_{i+1}$). The *jump number* of τ, $s(\tau, P)$ (resp. the *bump number* $b(\tau, P)$) equals the number of jumps (resp. bumps) of τ. The *jump number* $s(P)$ of P is the minimal number of jumps for which a linear extension of P can be found. Likewise the *bump number* $b(P)$ of P is the maximal number of bumps.

It is well known that for every linear extension τ of P, $s(\tau, P) + b(\tau, P) = |P| - 1$. So instead of minimizing the number of jumps, one can choose to maximize the number of bumps. This property shall be used later on, as it sometimes is more convenient to compute the number of bumps than the number of jumps. A linear extension with the minimum (maximum) number of jumps (bumps) will be called optimal.

Finding an optimal linear extension is called the *jump number problem.*

This problem has been first considered by Chein and Martin [4]. Pulleyblank [14] gave in 1981 the proof of its NP-completeness. This has also been done for specific classes of posets (e.g. with a chordal bipartite comparability graph by Müller [13]). For the case of interval orders, the jump number problem has recently been proved NP-complete by Mitas [10]. Specific classes of orders for which the jump number can be polynomially computed have been studied (for different surveys see [16, 2, 3]).

Let τ be a linear extension. $x_i x_{i+1} \ldots x_{i+k}$ is a *maximal chain* in τ iff

- $\forall j \in [i, i+k-1]$ x_j and x_{j+1} are separated by a bump,
- either $i = 1$ or x_{i-1} and x_i are separated by a jump,
- either $i + k = n$ or x_{i+k} and x_{i+k+1} are separated by a jump.

The *length* of a maximal chain is equal to the number k of elements it contains. We will call such a chain a *k-chain.*

A linear extension τ is said to be *k-chain* if it contains no maximal chain superior to k.

The *k-chain jump (bump) number* $s_k(P)$ $(b_k(P))$ is the minimal number of jumps (maximal number bumps) for which a k-chain extension exists.

2. Semi-orders and 3-chain linear extensions

For the rest of the paper we shall concentrate on maximizing the number of bumps. We first show that non-serializable semi-orders will offer good properties with respect to the k-chain measure. The following partitioning is classical:

Proposition 2.1. *Let $P = (X, \leqslant)$ be a poset. If P admits a series decomposition, i.e. X can be partitioned into two non-empty subsets X_1 and X_2 (yielding resp., P_1 and P_2) such that $\forall x \in X_1$, $\forall y \in X_2$, $x < y$, then $b(P) = b(P_1) + b(P_2) + 1$.*

The proof is straightforward. It is also known that such a decomposition requires only linear time [17].

A poset that cannot be partitioned will be called *non-serializable* (*n.s.*).

The serialization preserves (via Proposition 2.1) $b(P)$; therefore all posets will be required to be non-serializable (n.s.) in the sequel.

Lemma 2.2. *If P is an n.s. semi-order then for all x and y in P, $(x \ll y)$, xPy, Py and xP are n.s. semi-orders.*

Proof. Obviously all three are semi-orders. Suppose we can serialize xPy into xPz and $z'Py$, then every element w not in xPy is such that either $w < z$ or $z' < w$, so P would also be serializable. $\square$

We will next prove that n.s. semi-orders admit only 3-chain linear extensions.

Proposition 2.3. *Let P be an n.s. semi-order, then all chains in any linear extension of P are of length at most* 3.

Proof. Let τ be a linear extension of P, and $x_1x_2x_3x_4$ be a chain in τ. Take $y \parallel x_2$ and $y \parallel x_3$. P is a semi-order so $a_{x_2} < a_y$ and $b_y < b_{x_3}$. This would lead y to be after x_1 and before x_4 in all linear extensions, so such a y cannot exist. This induces that a serialization between x_2 and x_3 can take place, which contradicts the fact that P is non-serializable. $\square$

Let τ be a linear extension for an n.-s. semi-order P and denote the middle points of the maximal chains with length 3 by $c_1, \ldots, c_k$ so we can write $\tau = \tau_1 c_1 \tau_2 c_2 \ldots c_k \tau_{k+1}$. The sublinear extensions $c_i \tau_{i+1} c_{i+1}$ all have maximal chain length 2. In this section we show how you can use this observation. We shall characterize those elements of a semi-order which can be middle points of a 3-chain, and we shall show which of them can possibly occur together in one linear extensions. Furthermore, we show how these sublinear extensions of the form $c\tau'c'$ with maximal chain length 2 can be stuck together to give a bump number optimal linear extension. We first need some additional definitions.

Let P be a semi-order, x an element of P and τ a linear extension of P, x is a *centre in* τ if x is separated in τ to its left and to its right by a bump. A centre c in a linear extension has the property of splitting P in a unique way into two subsets, the intervals ending before c and the intervals starting after c. This is particular to semi-orders, and is not true on interval orders.

Lemma 2.4. *Let P be a semi-order and τ a linear extension of P with c as a centre.* $x \ll c \Leftrightarrow x <_\tau c$.

Proof. $\Rightarrow c$ is followed in τ by z such that $c < z \Rightarrow x < z \Rightarrow x <_\tau c$. $\Leftarrow$ for symmetrical reasons. $\square$

Let P be a semi-order, x is a *potential centre* if there exists a linear extension τ of P in which x is a centre.

Proposition 2.5. *Let P be a semi-order, x an element of P. x is a potential centre iff* $\mathrm{imsucc}(x)$ *and* $\mathrm{impred}(x)$ *are not void, and there exists no y such that* $\mathrm{succ}(x) \subset \mathrm{succ}(y)$ *and* $\mathrm{pred}(x) \subset \mathrm{pred}(y)$.

The construction of a valid linear extension is immediate.

Corollary 2.6. *Testing whether x is a potential centre can be done polynomially.*

Let P be a semi-order, x and y two potential centres of P are compatible if there exists a linear extension accepting them both as centres.

By Proposition 2.5 if x and y are centres then necessarily $x \ll y$ or $y \ll x$.

Proposition 2.7. *Let P be a semi-order, x and y two potential centres of P, $x \ll y$. Then x and y are compatible iff $A = \{z \ll y\} \cap \text{imsucc}(x) \neq \emptyset$, $B = \{z \gg x\} \cap \text{impred}(y) \neq \emptyset$ and $|A \cup B| \geq 2$.*

Proof. $\Rightarrow$ Let τ be a linear extension with $x_l x x_r$ and $y_l y y_r$ as maximal chains in τ. $x \ll y$. Then $x_r \in \text{imsucc}(x)$ because of x_1 and $x_r \ll y$ because of y_l. For the same reasons $y_1 \in \text{impred}(y)$ and $x \ll y_l$. Now $x_r \neq y_l$, $x_r \neq y$ and $x \neq y_l$, because we would have a 4- or 5-chain. Therefore $|A \cup B| \geq 2$.

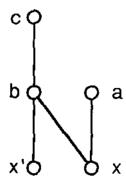
$\Leftarrow$ The third condition ensures that we can choose an element a from A, and an element b from B. Furthermore, by the requirements on A and B and Proposition 2.5, we have $a \ll b$, x is a potential centre for Pa, and y a potential centre for bP. So we can compute a linear extension of Pa with x as centre, followed by a linear extension of $P - (Pa \cup bP)$, again followed by a linear extension of bP with y as a centre, so that by Lemma 2.2 we finally obtain a linear extension of P with x and y as centres. □

Corollary 2.8. *Testing whether x and y are compatible centres can be done polynomially.*

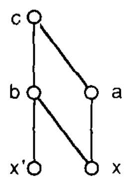
Proof. Constructing A, B and $A \cup B$ requires linear time, and this has to be done for all couples of potential centres. □

Proposition 2.9. *Let τ be a linear extension of xPy with maximal chain length 2. Then there is a linear extension τ' with maximal chain length 2, starting with x, and ending with y having at least the number of bumps as τ.*

Proof. It suffices to show that one can start with x since the proof of how to end with y is symmetric. Let τ be a linear 2-chain extension with x as early as possible. If τ starts with x we are done, so suppose x' is the predecessor of x in τ and τ looks locally like $x'xabc$. If xa is no bump we can exchange x and x' to get a contradiction of the minimality of x. Therefore we assume from now on that xa is a bump. Since the maximal chain length is 2 this implies that ab is no bump. If $x'a$ is a bump we get again the contradiction by exchanging x and x' so we assume all together $x < a$, $a \parallel b$, and $x' \parallel a$. In case $x'b$ is no bump we can exchange xa with x' to get $xax'bc$ which once more leads to the contradiction. Otherwise $x'b$ is also a bump and if $xax'bc$ has maximal chain length 3 we must have a further bump namely bc. Hence the order is locally depicted by the following Hasse diagram:



But since P is a semi-order either $x' < a$ which we excluded already or $a < c$. That is, P looks like



and $xx'bac$ gives the contradiction as well.

If y is one of the five elements there must be two distinct elements: one greater than x and one smaller than y because x and y are compatible centres. Hence $y = c$. □

From this it appears that an optimal linear extension can be found by looking through the different sets of compatible centres, finding the 2-chain number between all consecutive pairs and keeping the "best" subset. This can be dealt with in the following way.

Let $G(V, E, w)$ be the following weighted graph:

$$V = \text{set of potential centres of } P \cup \{0, 1\},$$

$$E = \{(c_1, c_2) \mid c_1 \ll c_2 \text{ and } c_1 \text{ and } c_2 \text{ are compatible centres}\}$$
$$\cup \{(0, c), (c, 1) \mid c \in V\} \cup \{(0, 1)\},$$

$$w(0,1) = b_2(P), \qquad w(0,c) = b_2(Pc), \qquad w(c,1) = b_2(cP),$$

$$w(c_1, c_2) = b_2(c_1 P c_2).$$

Theorem 2.10. *The maximal bump number of P equals the length of the longest $(0,1)$-path in G.*

Proof. $b(P) \geqslant$ longest (0,1)-path.

By Proposition 2.9 for all potential centres there is an optimal 2-chain extension of $c_1 P c_2$, $c_1 P$ and $P c_2$ starting with c_1 and ending with c_2. So the longest path induces a linear extension with length many bumps.

$b(P) \leqslant$ longest (0, 1)-path.

Let $\tau = x_1 \ldots x_{k_1} c_1 x_{k_1+1} \ldots x_{k_2} c_2 \ldots x_{k_n} c_1 x_{k_n+1} \ldots x_{k_{n+1}}$ be a bump number optimal linear extension. Then by Lemma 2.2 any subsequence of τ between two centres is a linear extension of the restricted order $c_i P c_{i+1}$ with maximal chain length 2. Therefore we can conclude that the number of bumps of τ between any two centres c_i and c_{i+1} is at most $b_2(c_i P c_{i+1})$. With the same argument the above inequality is valid for the beginning and the end of τ. Hence we have a path in G of length at least $b_2(P)$. □

Corollary 2.11. *Let P be an n.s. semi-order, $b(P)$ $[s(P)]$ is polynomially tractable if $b_2(P)$ is polynomially tractable.*

This result is a direct consequence of Proposition 2.1 and Theorem 2.10. The computation of the longest path is polynomial as our graph is acyclic. Alternatively, one may use the weigths $w(c_i, c_j) := |c_i P c_j| - 1 - b_2(c_i P c_j) = s_2(c_i P c_j)$ and then use the Dijkstra algorithm [5] to compute the shortest path in this weighted graph.

3. Computing the 2-chain number of semi-orders

The purpose of this section is to provide a polynomial algorithm which either computes $b_2(P)$ or says $b_2(P) < b(P)$. For this we shall associate with a given n.s. semi-order a special directed graph, called the bump graph of P. It shall be denoted $B(P)$. By Fishburn's [6] theorem, we can index the elements of Pred and Succ such that $\emptyset \subset \text{pred}_1 \subset \text{pred}_2 \subset \cdots \subset \text{pred}_m$ and $\text{succ}_1 \supset \text{succ}_2 \supset \cdots \supset \text{succ}_m \supset \emptyset$ where the letter m associated to the poset is of course equal to $|\text{Pred}| - 1$. This induces an order on the set $Y = \text{Pred} \cup \text{Succ} - \{\emptyset\}$ by numbering $y_1 \ldots y_{2m}$, with the following rule.

$\forall i \in [1, m]$, $y_{2i-1} = \text{succ}_i$, $y_{2i} = \text{pred}_i$. The bump graph $B(P) = (V, U)$ is now defined as follows:

$$V := X \cup Y = X \cup \text{Pred} \cup \text{Succ} - \{\emptyset\},$$

$$\begin{aligned} U := {} & \{(x y_i) \mid x \text{ has as successor set } y_i\} \\ & \cup \{(y_i x) \mid x \text{ has as predecessor set } y_i\} \\ & \cup \{(y_i y_{i+1}) \mid i \in [1, 2m-1]\}. \end{aligned}$$

The so defined graph $B(P)$ is unique and acyclic. We recall that a matching on $B(P)$ is a subset W of U such that no two edges of $B(P)$ have a same common endpoint. A matching is maximal if there is no matching W' with $W \subset W'$. A matching is called maximum if there is no matching W' with $|W'| > |W|$. It is also well-known [1] that finding a maximum matching is a polynomial problem. This can be dealt with for example by the Micali and Vazirani algorithm [10].

Proposition 3.1. *For every linear extension τ of an n.s. semi-order P, there exists a maximal matching W on $B(P)$ of size $m + b_2(\tau, P)$.*

Proof. For every maximal chain of length 2, xx' add to W edges xy_j and $y_k x'$ where y_j corresponds to succ(x), y_k to pred(x'). This implies $|W| = 2b_2(\tau, P)$. The number of elements of Y endpoints of such edges is also $2b_2(\tau, P)$.

Now if y_j and y_k are endpoints and $\forall i \in [j+1, k-1]$, y_i is not an endpoint, then $k - j$ is odd. To prove this we should note that:

Case 1: j and k are odd ($y_j = \text{succ}(x)$, $y_k = \text{succ}(x')$) and there are two bumps xz and $x'z'$ in τ. But since no y_i between y_j and y_k is matched we must have $x < z'$, $x' < z$ so x' should be computed in τ before z and x before z', which is impossible.

Case 2: This is symmetrical with j and k even.

For identical reasons the smallest i such that y_i is endpoint is odd. Therefore, in Y, between two consecutive endpoints there is an even number of elements of Y. A basic pairwise matching is possible such that all elements of Y are endpoints, which assures us that W is now maximal. We have added $(m - b_2(\tau, P))$ new edges so the total of edges in W is $b_2(\tau, P) + m$. □

Proposition 3.2. *For every maximum W matching of $B(P)$ of size n, we can construct a linear extension τ of P with at least $n - m$ bumps.*

Proof. The construction will be made in three steps. The first two involve transformations of W, preserving the maximum matching, but changing some edges to finish with a graph equivalent to one we could have found through the application of Proposition 3.1, the last step is an algorithm that effectively computes the linear extension.

Step 1: We transform W into an equivalent matching such that every y_i is endpoint of an edge in W. To do this consider the smallest i such that y_i is not endpoint of an edge in W. Necessarily $i \neq 2m$ since the matching would then be completed by $y_i x$, for $y_i = \text{pred}(x)$. Then y_{i+1} is endpoint (if not there exists a larger matching by adding $y_i y_{i+1}$ to W), so replace the edge with endpoint y_{i+1} by the edge $y_i y_{i+1}$ giving a matching of same size, so also maximum.

Iterate this procedure until every y_i is endpoint of an edge in W. As the end of the matching is still maximum, and every y_i is endpoint of some edge in the matching.

Step 2: It is easy to see that W is in the same form as the W obtained by Proposition 3.1's proof. Nevertheless, for a given couple $y_j y_k$ such that $\exists x, x' \in X$ such that xy_j and $y_k x'$ belong to W with all intermediate vertexes of Y related 2 by 2, we can have $x < x'$ but not x' covering x. W can be transformed to deal with this problem: for every such couple $y_j y_k$, choose a successor z of x s.t. i with $\text{pred}(z) = y_i$ is minimal. Then replace in W edges $y_q y_{q+1}$ (for all q in $[i, k-2] \cap 2\mathbb{N}$) and edge $y_k x'$, by edges $y_{q+1} y_{q+2}$ (for all q in $[i, k-2] \cap 2\mathbb{N}$) and edge $y_i z$. W remains a matching, of identical size, hence maximum, and the step 1 property still holds.

Step 3: The linear extension τ will be computed by the following algorithm. Its main idea is: at any one moment, add to the linear extension the free elements of X, i.e. those whose predecessors have already been computed and which are not endpoints in the maximum matching W; if there is no free variable then add a bump, i.e. xx' such that xy_i and $y_j x'$ are the earliest edges in W. The algorithm takes as entry $B(P)$ and deletes step after step vertices from $B(P)$ as well as adjacent edges to free elements.

Algorithm.

Procedure Free($B(P)$);
begin
for all x in X do
  if indegree(x) $= 0$ and x not matched then add x to τ
end;

main
(0) Free($B(P)$);
(1) $i := 1$;
(2) while $i < 2m$ do
      begin
(3)   if $y_i y_{i+1}$ in W then
        begin
(4)   remove y_i, y_{i+1} and their adjacent edges from $B(P)$;
(5)   Free($B(P)$);
(6) $i := i + 2$;
(7)   end
(8)   else
        begin
(9)   add x with $xy_i \in W$ to τ;
(10) $k := \min\{j > i \mid \exists x', y_j x' \in W\}$;
(11)   add x' to τ;
(12)   remove all vertices y_j with $j \in [i, k]$ and their adjacent edges from $B(P)$;
(13)   Free($B(P)$);
(14) $i := k + 1$;
(15)   end;
(16) end;

Proof of the algorithm. Observe that i in line 3 is always odd, so y_i is a successor set and for any predecessor z of x with $\mathrm{pred}(z) = y_j$ and $\mathrm{succ}(z) = y_k$ we have $j < k < i$. Hence all predecessors of x are added to τ already by a preceeding execution of line 9 or 11, or Free($B(P)$). By the modification of the matching in step 2 we have insured that x' can be added after x.

Since for any pair $y_j y_k$ such that xy_j and $y_k x'$ is in W with all intermediate vertices matched 2 by 2 we created one bump there are at least $n - m$ bumps. But there might be more bumps in τ, because by adding an element from Free($B(P)$) we might add a 3-chain and hence an extra bump. □

Theorem 3.3. *The bump number problem for n.s. semi-orders is polynomial.*

Proof. For any edge (x, y) of the graph in Theorem 2.10 we construct the bump graph for the associated order xPy, calculate a maximum matching of size n, and with the algorithm of Proposition 3.2 a linear extension τ with at least $m - n$ bumps where $m = |\text{Pred}| - 1$. If τ contains a 3-chain then $b(\tau, xPy) > m - n$ and there is a path from x to y. By Proposition 3.1 this path is longer than the edge xy. Hence the edge xy cannot be contained in a longest path since G is acyclic. Therefore we can remove the edge xy from G without changing the longest path. Otherwise if τ does not contain a 3-chain it is an optimal 2-chain extension and we can use $b_2(\tau)$ as weight for this edge. Solving the longest path problem in the remaining graph and using the extensions on this path produces an optimal linear extension. □

4. Example and complexity issues

Let P be the semi-order represented by intervals in Fig. 3. P can be serialized between d and e into P_1 and P_2.

In P_1 there are no potential centres; in P_2 these are h, i, j and k. The compatibility test yields that the labelled graph (for P_2) which has to be considered before labelling is the one shown in Fig. 4. We then label this graph via the computations of the different $b_2(P)$. Take, for example Pk, the associated bump graph $B(Pk)$ which is represented by Fig. 5, with $m = 4$.

A maximum matching of size 7 is obtained in Fig. 6, yielding through Proposition 3.2 linear extension e, h, f, i, g, k, j with 3 ($= 7 - 4$) maximal chain of length at least 2. We therefore obtain the graph of Fig. 7, on which a longest path is obtained going through vertex j. The number $b(P_2)$ is therefore 5 which added to $b(P_1) = b_2(P_1) = 2$, plus one (see Proposition 2.1) gives 8, respected by the linear extension

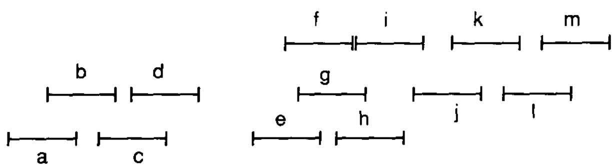


Fig. 3.

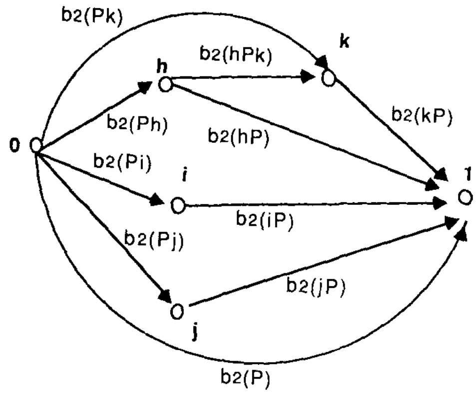


Fig. 4.

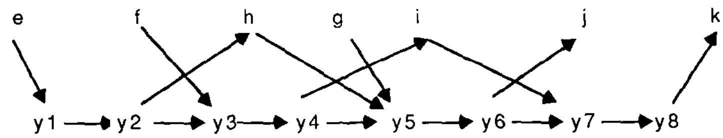


Fig. 5.

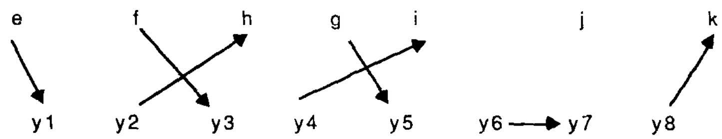


Fig. 6.

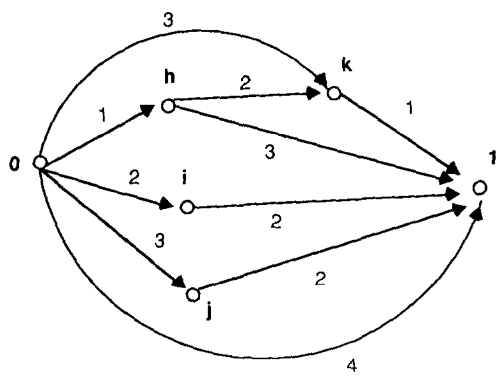


Fig. 7.

$a, c, b, d, e, h, f, i, g, j, l, k, m$. This of course yields a jump number $s(P) = |P| - b(P) - 1 = 13 - 8 - 1 = 4$.

4.1. Some complexity considerations

We give hereafter an estimation of the time cost for the effective computation of $s(P)$, where $|P| = n$. These estimations are based on a certain amount of well-known algorithms. Alternative algorithms and data structures have not been considered. Interval representation can be obtained in $O(n)$ [11]. Construction of the different sets impred(x), imsucc(x), Pred and Succ can take place in $O(n)$. Serialization can be done in $O(n)$ [17]. Construction of the set of potential centres(P) is linear. Construction of the auxiliary graph for labelling is in $O(n^2)$. Computation of the longest path in an acyclic graph is in $O(n^3)$.

Computation of $b_2(P)$ costs: $O(n)$ for the construction of the bump graph. $O(n^{3/2})$ for the computation of the size of the maximum matching by the Micali and Vazirani algorithm [9]. (The number of vertexes of the bump graph is bounded by $3*n$, the number of edges by $4*n$). This has to be done for every couple of compatible centres, for every centre twice $(b_2(Pc), b_2(cP))$ and for $(b_2(P))$. This yields $O(n^{3.5})$.

The overall complexity is therefore $O(n^{3.5})$. To obtain the complexity of the actual computation of an optimal linear extension we must compare the above complexity with the complexity of the algorithm given in the proof of Proposition 3.2, since we have to compute the linear extension only when the shortest path is discovered. Its complexity being below $O(n^{3.5})$, this is the complexity also for the computation of an optimal linear extension.

This complexity can certainly be lowered, but it has not been done yet.

References

[1] C. Berge, in: Dunod ed., Graphes et Hypergraphes (1973) 117–121.

[2] V. Bouchitté and M. Habib, Some NP-completeness properties about linear extensions, Order 4 (1987) 143–154.

[3] V. Bouchitté and M. Habib, The calculation of invariants for ordered sets, in: I. Rival, ed., Algorithms and Order (Kluwer, Dordrecht, 1989) 231–279.

[4] M. Chein and P. Martin, Sur le nombre de sauts d'une forêt, C.R. Acad. Sciences Paris, t. 275, Série A (1972) 159–161.

[5] E. Dijkstra (1958), A note on two problems in connection with graphs, Numer. Math. 1 (1959) 269–271.

[6] P.C. Fishburn, Intransitive indifference in precedence theory: A survey, Per. Res. 18 (1970) 207–228.

[7] M.C. Golumbic, Interval graphs and related topics, Discrete Math. 55 (1985) 113–121.

[8] M. Habib, M. Morvan and J.-X. Rampon, Remarks on some concurrency measures, in: Lecture Notes in Computer Science (Springer, Berlin, 1990), to appear.

[9] S. Micali and V.V. Vazirani, An $O(V^{1/2}E)$ algorithm for finding maximum matching in general graphs, in: 21st Annual Symposium on foundations of Computer Science, Syracuse, 1980 (IEEE Computer Society Press, New York, 1980) 17–27.

[10] J. Mitas, Tackling the Jump Number of Interval Orders, Order 8 (1991) 115–132.

[11] R.H. Möhring, Computationally tractable classes of ordered sets, in: I. Rival, ed., Algorithms and Order (Kluwer, Dordretch, 1989) 105–194.
[12] B. Monjardet, Axiomatique et propriétés des quasi-ordres, Math. Sci. Hum. 63 (1978) 51–82.
[13] H. Müller, Alternating cycle-free matching, Order 7(1) (1990) 11–22.
[14] W.R. Pulleyblank, On minimizing setups in precedence constrained scheduling, Preprint, 1981.
[15] I. Rabinovitch, The dimension of semi-orders, J. Combin. Theory (A) 25 (1978) 50–61.
[16] M.M. Syslo, A graph theoretic approach to the jump number problem, in: I. Rival, ed., Graphs and Order (Reidel, Dordrecht, 1985) 185–215.
[17] J. Valdes, R.E. Tarjan and E.L. Lawler, The recognition of series-parallel digraphs, Proceedings of the 11th Annual ACM Symposium on Theory of Computing (1979) 1–12.